

# Interactive 3-D Graphics for the Apple II

*An understanding of the theory of perspective  
enables you to represent three-dimensional objects  
on a two-dimensional screen.*

---

Andrew Pickholtz  
3613 Glenbrook Rd.  
Fairfax, VA 22031

---

In the present generation of computers, no other form of output rivals the popularity of the video terminal with its two-dimensional visual representation of data. This article will examine ways of making this two-dimensional output represent the three-dimensional real world. Techniques of showing perspective play an important role in making video output look three-dimensional. In this article, I will look briefly at the concept of perspective and then consider some techniques of achieving perspective in computer graphics. I will then present some program listings in

BASIC and Pascal that show how to use these techniques in high-level languages.

## Ways of Representing Three Dimensions

People tried to portray the visual world on a flat screen long before the creation of the modern computer, and draftsmen today use several dif-

---

## A computer can as easily produce a perspective drawing as an oblique or isometric drawing.

---

ferent methods of representing three-dimensional objects on paper: the orthogonal, the oblique, the isometric, and the perspective methods.

An orthogonal projection of an object is simply the "side view" of that object (see figure 1). "Side view" is in quotes because, as will later become clear, this representation is not exactly what the human eye would see if it were looking at the object; that is, this "view" is not a perspective projection.

Orthogonal representations of ob-

jects customarily give three projections: one from the top, one from the front, and one from the right-hand side. Each "view" gives information about a pair of axes; the "top view" gives information about the  $x$ - $y$  pair, the "front view" about the  $x$ - $z$  pair, and the "right-hand side view" about the  $y$ - $z$  pair. Unfortunately, the untrained eye is reluctant to form a three-dimensional image from the three detached and seemingly independent illustrations used in orthogonal representation.

Oblique and isometric drawings (see figures 2 and 3, respectively) portray an object in a more realistic manner. Both the oblique and isometric representations depict a three-dimensional object in one illustration by fixing the axes in relation to the horizontal. In oblique pictorial, lines parallel to the  $z$  axis are vertical, lines parallel to the  $x$  axis are horizontal, and lines parallel to the  $y$  axis are consistently drawn at the same angle in relation to the horizontal. The axes in isometric pictorial are likewise fixed in relation to the paper.

## The Perspective Method

While oblique and isometric representations are superior to orthogonal,

---

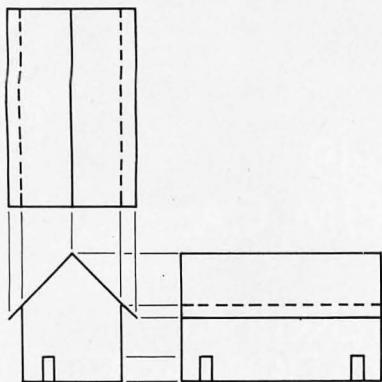
### About the Author

Andrew Pickholtz wrote this article while a senior at W. T. Woodson High School in Fairfax, Virginia. He is now a student at Harvard University. In summers, he has worked for Ferox Microsystems Inc. and at the IBM T. J. Watson Research Center.

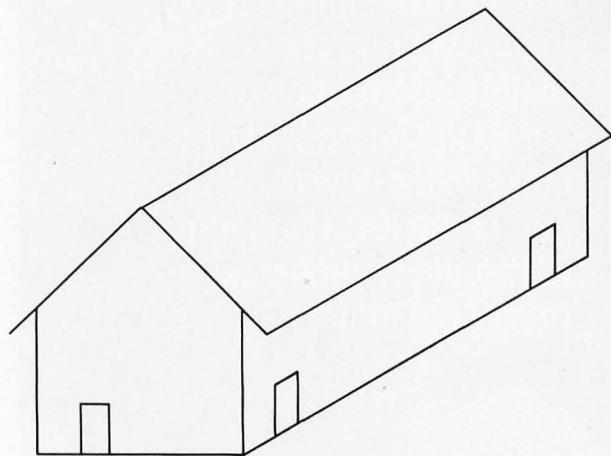
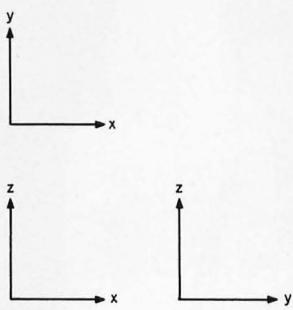
---

### Acknowledgments

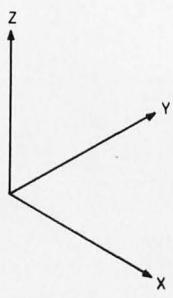
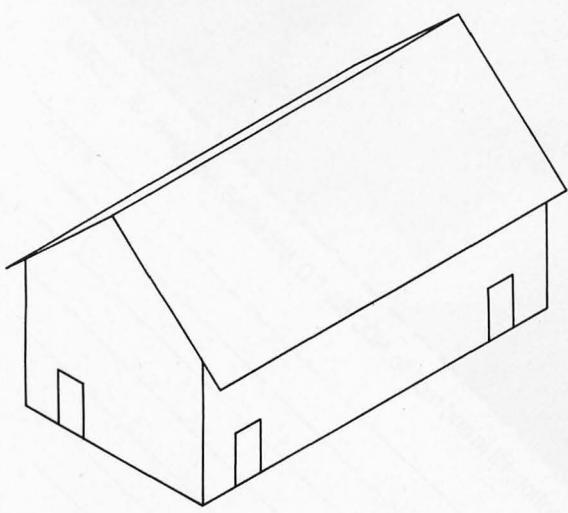
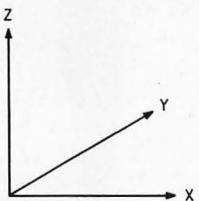
The author would like to thank the Department of Electrical Engineering and Computer Science, George Washington University, for the use of its word processor and Apple II computer systems. He would also like to thank Dr. and Mrs. R. Pickholtz for persuading him to write this article.



**Figure 1:** An orthogonal representation of a house. An orthogonal drawing drops perpendiculars from each point on the object to three mutually perpendicular planes. Hidden edges are customarily drawn as dotted lines.



**Figure 2:** An oblique representation of a house. An oblique drawing portrays three dimensions by drawing lines parallel to the third axis at a consistent angle to the horizontal, in this case at 30 degrees.



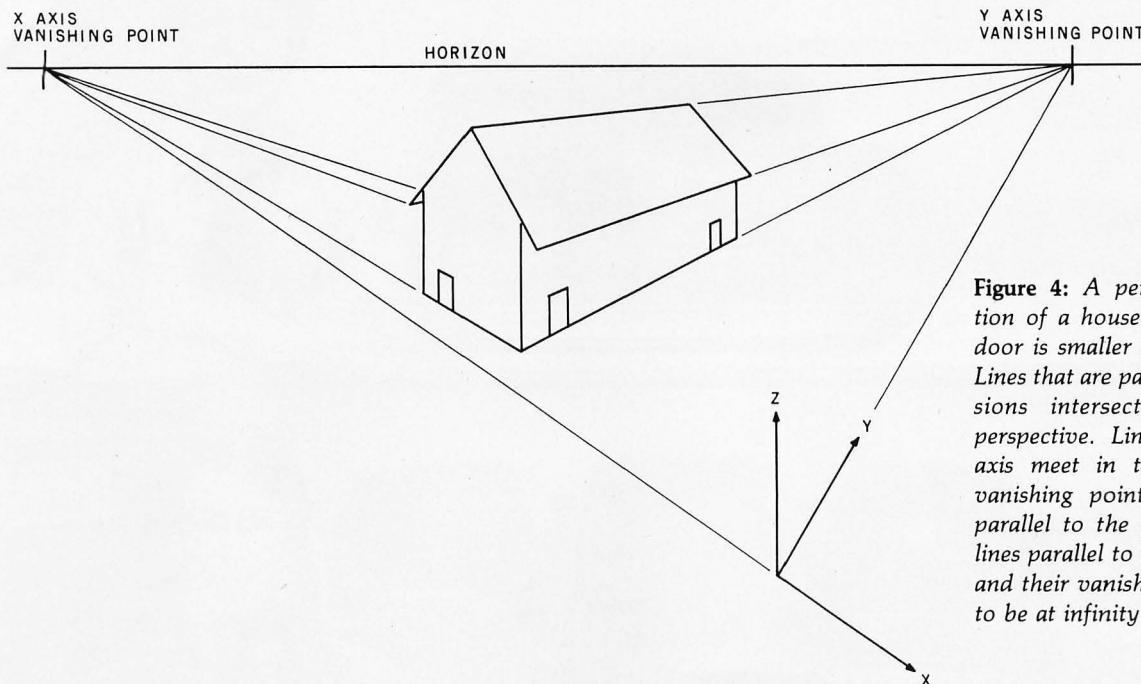
**Figure 3:** An isometric representation of a house. Like an oblique representation, an isometric one draws lines that are parallel in three dimensions as parallels in two. The isometric method, however, offsets two axes from the horizontal.

perspective pictorial is the only truly accurate method of illustrating an object. Two Florentine architects, Filippo Brunelleschi and Leon Battista Alberti, developed the ideas of perspective in the fifteenth century. Although many artists before them had noticed that objects in the distance appear smaller than objects in the foreground, Brunelleschi and Alberti were the first to accurately represent the apparent diminution of objects as they recede from the observer. Many other Italian artists and some Flemish artists had also experimented with perspective; however, their methods were empirical while Brunelleschi and Alberti worked with a geometric system. In fact, Alberti had written several papers on mathematics, and in 1435 wrote the first treatise on painting that dealt with the theory of art rather than just the techniques.

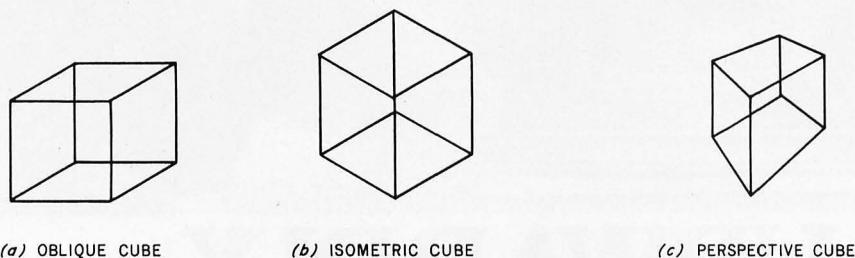
What makes perspective drawings superior to oblique and isometric is that perspective displays objects in the distance as smaller than objects that are closer; the rear door in figure 4, for example, is smaller than the front door. Perspective drawing also represents lines that are parallel in three dimensions as convergent on the picture plane. Thus, the axes in perspective drawings are always directed toward vanishing points. The x-axis and y-axis vanishing points in figure 4 lie on the horizon; an object in the distance would, as the eye expects, appear extremely small.

Figures 5a-5c illustrate another interesting fact about perspective: while the oblique (figure 5a) and isometric (figure 5b) representations of a wireframe cube appear to spontaneously reverse in orientation, the perspective representation (figure 5c) does not. What prevents the spontaneous reversal in the perspective representation is that one of the perceived orientations of the perspective cube is erroneous; that is, it does not look "natural."

Although oblique and isometric drawings are not truly realistic, draftsmen use these two techniques more often than perspective. They do this for two reasons. First, oblique and isometric drawings conveniently



**Figure 4:** A perspective representation of a house. Note that the rear door is smaller than the front door. Lines that are parallel in three dimensions intersect when drawn in perspective. Lines parallel to the x axis meet in the distance (at the vanishing point), and so do lines parallel to the y axis. In this case, lines parallel to the z axis are vertical and their vanishing point is assumed to be at infinity.



**Figure 5:** Three representations of a cube. Figure 5a is an oblique representation, figure 5b is isometric, and figure 5c is a perspective. Both the oblique and the isometric representations appear to reverse in orientation spontaneously. The perspective does not.

permit finding the measurements of an object by simply measuring the representation; second, drawing perspective is much more difficult. For a computer, however, it is just as easy to produce a perspective drawing as to produce an oblique or isometric drawing. Furthermore, as the object becomes more complex, the difference in speed between computer-drawn perspective and computer-drawn isometric becomes negligible.

### Describing a Three-Dimensional Object

It is impossible to produce a perspective pictorial of an object without a description of the object. A good representation of the object can usually be achieved by assuming that the object is composed of a finite

number of planar polygons. If the object is significantly curved, an adequate representation requires many polygons.

Figure 6 illustrates a data structure that describes a three-dimensional object. Each of the polygons, which can be called faces, is composed of edges. Each edge is composed of two vertices that are specified by three Cartesian coordinates. Each face also has several characteristics: color, texture, transmittance, glossiness, and reflectance. The edge shared by two faces is the intersection of their two sets of coordinates.

It is easier to represent an object if we assume that the object has clear faces. This simplification avoids the difficult problem of discovering hidden lines. Figure 7 shows the simpler data structure that this assumption

permits to represent the wire-frame object previously represented in figure 6.

### Specifying an Arbitrary Three-Dimensional View

We can think of a perspective pictorial of a three-dimensional scene as a view that a one-eyed pilot would see when looking through an empty picture frame (see figure 8). The picture frame is understood to lie in the picture plane. As the figure shows, the pilot's line of sight is defined to be the normal (perpendicular) to the picture plane that passes through the pilot's eye. The lines connecting the object with the pilot's eye are called projectors. The perspective pictorial is the intersection of the projectors and the picture plane.

Three general types of changes would affect the pilot's view of the scene: a change in the distance between the picture plane and the pilot's eye, a change in position of the aircraft, or a rotation of the airplane. If the picture plane is moved closer to the pilot's eye, the view would appear smaller in comparison to the picture frame. Likewise, if the picture frame is moved further away from the pilot's eye, the view would appear larger since the tetrahedral angle that the picture frame subtends (marks off) would be smaller. Thus, in order to specify any three-dimensional

VERTICES	$v_1(x_1, y_1, z_1)$	$v_2(x_2, y_2, z_2)$	... $v_{NV}(x_{NV}, y_{NV}, z_{NV})$
EDGES	$E_1 = v_1, v_4$	$E_2 = v_5, v_9$	$E_3 = v_2 v_{104} \dots E_{NE} = v_{16} v_3$
FACES	$F_1 = E_1 E_3 \dots E_{118}$	$F_2 = E_{209} E_5 E_{19} \dots F_{NF} = E_3 E_{52} \dots E_{129}$	
COLOR <sub>1</sub>	COLOR <sub>2</sub>	... COLOR <sub>NF</sub>	
TEXTURE <sub>1</sub>	TEXTURE <sub>2</sub>	... TEXTURE <sub>NF</sub>	
GLOSSINESS <sub>1</sub>	GLOSSINESS <sub>2</sub>	... GLOSSINESS <sub>NF</sub>	
REFLECTANCE <sub>1</sub>	REFLECTANCE <sub>2</sub>	... REFLECTANCE <sub>NF</sub>	
OBJECT	$O_1 = F_1 F_2 F_3 \dots F_{NF}$		

**Figure 6:** A data structure representing an object. The representation assumes that the object is composed of a finite number of polygons (also called faces). Each face has several characteristics (color, etc.) and is determined by the edges that it contains. Each edge is specified by its endpoints, which are the vertices of the object. Finally, the coordinates of each vertex must be specified. The object represented here is hypothetical.

VERTICES	$v_1(x_1, y_1, z_1)$	$v_2(x_2, y_2, z_2)$	... $v_{NV}(x_{NV}, y_{NV}, z_{NV})$
TRAILS	$T_1 = v_1 v_3 v_8 \dots v_{186}$		
	$T_2 = v_9 v_{12} v_{136} \dots v_1$		
	$T_3 = v_{13} v_{12} \dots v_{13}$		
	.		
	$T_{NT} = v_4 v_{11} \dots v_{22}$		
OBJECT	$O_1 = T_1 T_2 T_3 \dots T_{NT}$		

**Figure 7:** A data structure representing a wire-frame object. This data structure assumes that the object has transparent faces. The object is composed of trails. Each trail is defined by the vertices that it contains. Each vertex is specified by its coordinates. The data structure shown represents a hypothetical object.

view, we must know the distance between the picture plane and the observer's eye, the position of the observer, and the angular position of the observer's line of sight.

Figure 9 illustrates the linear and angular position of an observer. Three Cartesian coordinates specify the location of the observer. The coordinate axes that specify the observer's location are the same axes used to specify the vertices of the object. Describing a unique line of sight requires three angles—pitch, bank, and heading. A change in pitch is a rotation about the wings. A change in bank, or roll, is a rotation about the fuselage. And a change in heading, or yaw, is a rotation about a vertical line passing through the pilot; in other words, the heading is the compass direction of the airplane.

Since rotation is not a commutative operation—one in which a change in order will not change the results—we must declare an order of precedence for pitch, bank, and heading. The most physically appealing order is heading, pitch, and then bank. Using that order, we can determine a line of sight by first rotating a unit vector parallel to the  $y$  axis about the  $z$  axis in an amount specified by the heading. Next, we should rotate the new vector about the new position of the wings by an amount specified by the pitch. And finally, we should

## Communications Software!

**Why do people prefer E/LYNC to programs selling for five times as much?**

### E/LYNC      \$30

**E/LYNC is a powerful yet easy to use program with the features they want most:**

- \* Transfer ANY type of file from one computer to another
- \* Transfer files between any two computers running E/LYNC or LYNC
- \* Automatic error detection and correction when transferring files
- \* Messages typed in Conversation mode appear on both computers
- \* Real-time display of total bytes sent or received
- \* Connect to time-share systems using phone modem
- \* Upgrade to LYNC for \$95

**LICENSING:** E/LYNC & LYNC are licensed on a per user basis. If you intend to use E/LYNC or LYNC to communicate with another person, they must also purchase E/LYNC or LYNC. Versions for CP/M are written in Z-80 assembly language. Versions for IBM PC are written in IBM 8088 assembly language and require PC DOS.

**AVAILABLE DISK FORMATS:** 8" Standard, IBM PC DOS, 8" TRS-80 Mod II with CP/M, Vector Graphic, Cromenco, North Star, Osborne, Xerox 820 (8" or 5"), SuperBrain, NLS KayPro. Special TRS-80 Mod III version of LYNC available for \$75. Apple Version coming soon!

**TERMS:** Money Order, Check, C.O.D., MC or VISA, First Class shipping included. UPS Blue add \$2. C.O.D. add \$3. CA Res. add 6%. Overseas airmail add \$5. Manuals only, \$10 (specify CP/M or IBM PC).

### LYNC      \$125

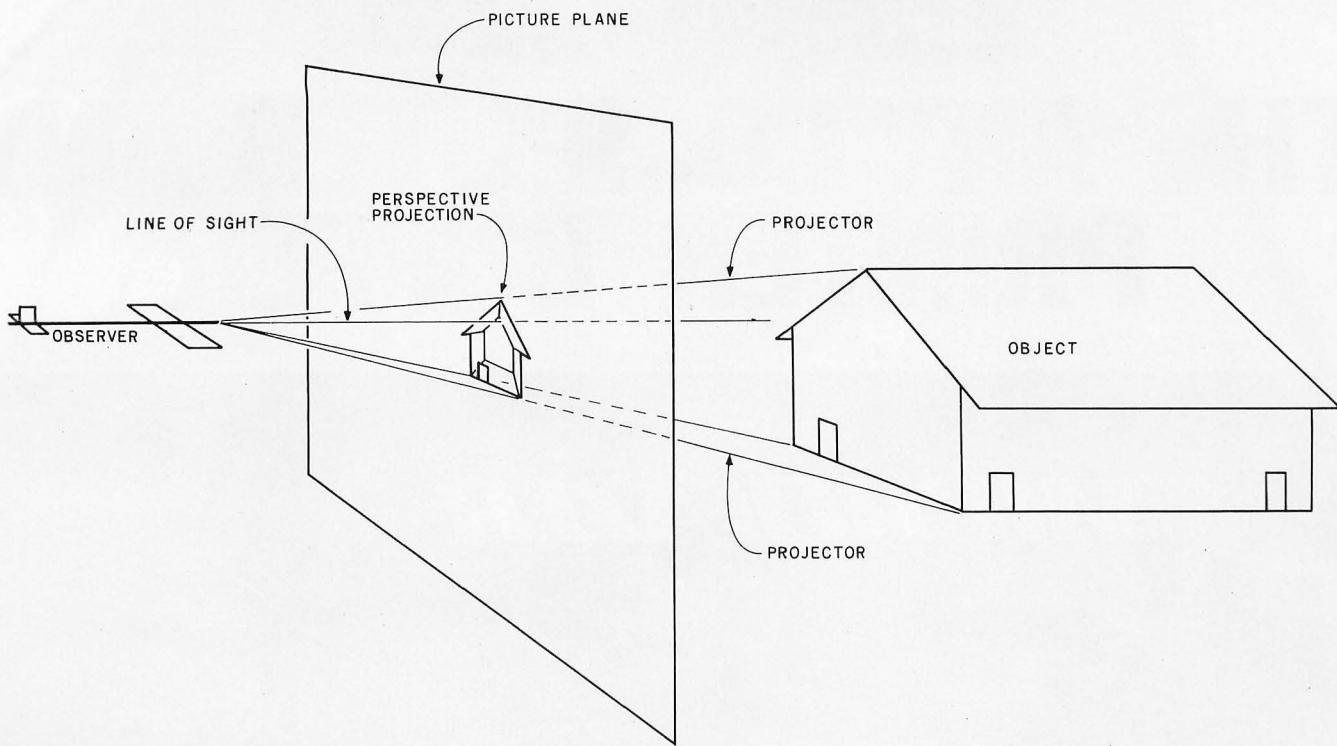
**Frequent or professional users will want the additional features of LYNC ...**

- \* Multiple file transfers including full 'wildcard' matches
- \* Fetch files from the remote computer
- \* View disk directory of BOTH local and remote computer
- \* Re-log onto another disk drive of local or remote computer
- \* Send to and Receive files from a time-sharing computer
- \* Optional XON/XOFF protocol

PHONE ORDER DESK: (805) 966-3077 10am-4pm PST Monday-Friday

**INTERNATIONAL SOFTWARE ALLIANCE  
1835 MISSION RIDGE ROAD SANTA BARBARA, CA 93103**

E/LYNC & LYNC are trademarks of Midnight Software    CP/M is a trademark of Digital Research



**Figure 8:** Perspective projection of an object. The observer's line of sight is normal to the picture plane. The projectors of an object are the lines connecting the object to the observer's eye. The perspective projection consists of the intersections of the projectors and the picture plane. The distance between the observer and the picture plane controls the size of the perspective projection; the farther the picture plane is from the observer, the larger the projection.

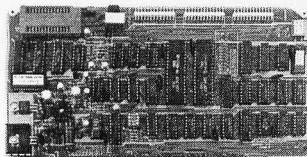
rotate this new vector about the newly positioned fuselage in an amount specified by the bank.

### Solving for the Standard Position

Later, we will see that the computations required to create a perspective projection can be greatly simplified by translating and rotating the coordinate system so that the observer is at the origin, with the line of sight aligned with the positive  $y$  axis. When the observer is in this standard position, the pitch, bank, and heading are defined to be zero. We can move the observer to the standard position only if we likewise move the three-dimensional scene so that the observer's view remains unchanged.

Assume that the observer is at the location  $(X_v, Y_v, Z_v)$  and has pitch, bank, and heading  $p$ ,  $b$ , and  $h$ , respectively. Translating the observer to  $(0,0,0)$  and a point  $Q(X, Y, Z)$  to  $(X - X_v, Y - Y_v, Z - Z_v)$  does not alter

## EXTENDED PROCESSING S-100 BOARDS



**BURNER I/O**

The BURNER I/O has a complete EPROM programmer, two serial ports, one parallel I/O port with handshaking and memory management.

#### Programmer features:

- Programs EPROM types 2704 thru 2764, 2508, 2516 and TMS2716.
- CP/M compatible software supplied in EPROM that can be easily written on a diskette.
- Programming socket is zero insertion force type.
- Programming voltages generated on board.
- Programmer is totally I/O mapped.

#### I/O features:

- 2 fully independent RS-232 serial ports with data ready.
- Independent baud rate generators are software programmable from 50 to 19,200 baud.
- Serial ports may be polled and/or interrupt driven.
- Independent parallel latched 8 bit output, input and status flags.
- There are 4 unlatched input bits.

Memory management controls the S-100 address lines from A16-A23.

This board is offered with all options, or just the portions that are needed. Regardless of the version purchased, documentation for the entire board is supplied. All combinations are assembled and tested.

Option A: Complete board with programmer, I/O and memory management. \$324.95

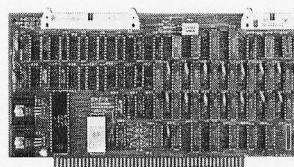
Option B: Programmer only, \$199.95

Option C: I/O only, (25 + P) \$199.95

Option D: Option B and C, \$299.95

Option E: Memory management only \$99.95

Memory management may be added to options B or C for \$25.00. 10% price increase effective 1/1/83.



**EPZ SINGLE BOARD COMPUTER**

The EPZ is a single board S-100 computer designed to run in parallel with your existing computer. It is also useful as a stand alone computer for controlling equipment outside of the S-100 system. The host

computer can communicate with it through the S-100 bus via a built in parallel port. It has its own ROM monitor that is designed to operate with the host computer.

#### Features include:

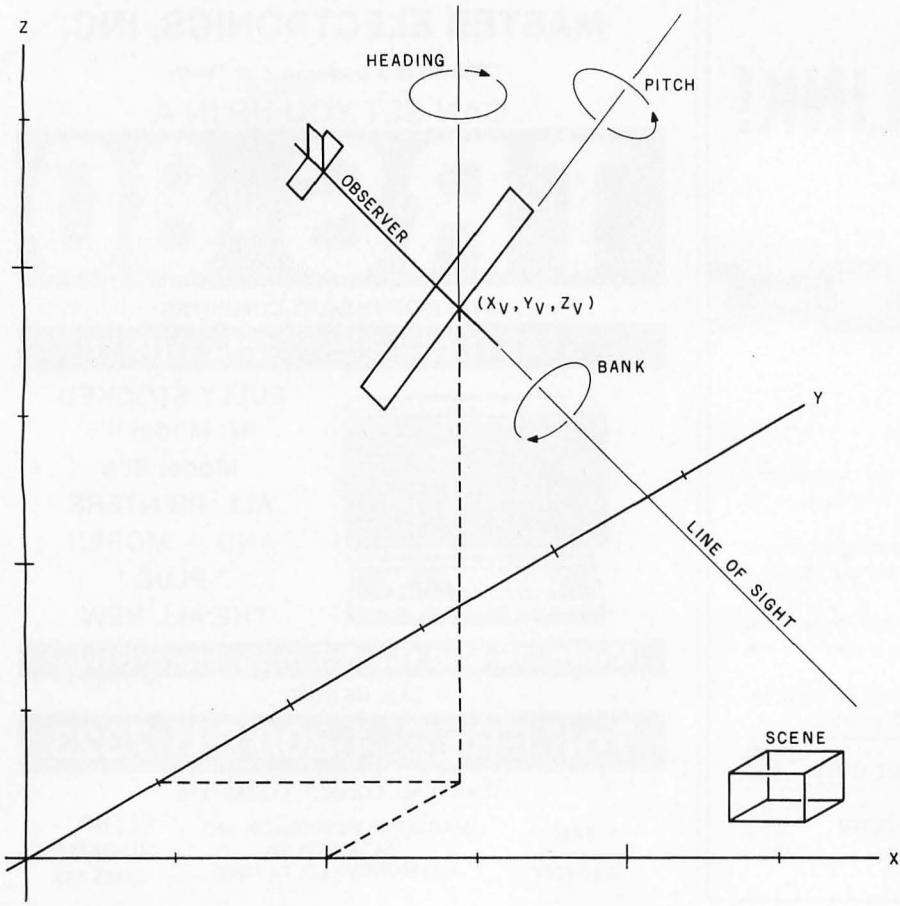
- 4MHz Z80A CPU.
- ROM operating system.
- 8K of fast static RAM.
- 2K EPROM (2716) expandable to 4K (2732).
- Software supplied to interface 8080/Z80 host CPU.
- 8 bit parallel I/O and status flags for user applications.
- Buffered bus on 50 pin connector for special controller applications. All Z80 signals are available.
- Top quality construction including sockets and 4 layer construction for operation in the noisiest of environments.
- Optional CP/M based utility software for easier EPZ access.
- As many EPZ's as desired may operate in one host computer.
- Price of \$495.95

#### AVAILABLE FROM

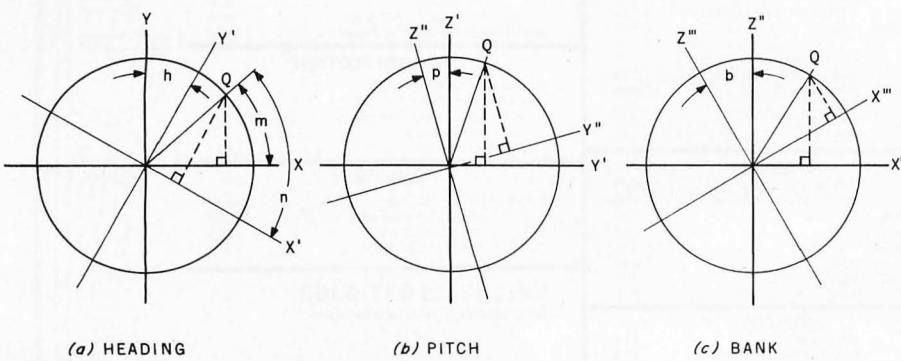
**Microbyte Computer Systems**

2798 So. Bascom, San Jose, CA 95124  
(408) 377-4685

**DEALER INQUIRIES INVITED**



**Figure 9:** Viewing parameters. In order to specify a unique view of an object in three dimensions, it is necessary to declare the observer's location and the angular position of the line of sight. The observer is at the point  $(X_v, Y_v, Z_v)$ , and the line of sight is specified by the aircraft's pitch, bank, and heading.



**Figure 10:** Rotational transformation of a point. These figures illustrate the relationships between the coordinates of a point  $Q$  and the coordinates of  $Q$  in a Cartesian system where the observer is in the standard position. The standard position occurs when the observer is located at the origin, the line of sight is the positive  $y$  axis, and the "wings" lie on the  $x$  axis. In this position, the observer's pitch, bank, and heading are defined to be zero. See equations (1) through (9) in the text.

the observer's view of the point, but simplifies the rotations that follow. Furthermore, a rotation of the coordinate axes will not affect the observer's perception of the point  $Q$  if the point's coordinates undergo an appropriate rotational transformation.

Since three rotations from the standard position determine a line of sight, three rotations of the coordinate axes are needed to bring the observer into the standard position of the rotated coordinate system. Again, the order of rotation is important. First, the  $x$  and  $y$  axes are rotated about the  $z$  axis by the amount of the heading,  $h$ , so that the airplane's fuselage is in the  $y'z'$  plane (zero heading in the  $x'y'z'$  system). Next, the  $y'$  and  $z'$  axes are rotated about the  $x'$  axis by the amount of the pitch,  $p$ , so that the fuselage lies on the  $y''$  axis (zero heading and zero pitch in the  $x''y''z''$  system). Finally, the  $x''$  and  $z''$  axes are rotated about the  $y''$  axis by the amount of the bank,  $b$ , so that the pilot is in the standard position in the  $x'''y'''z'''$  coordinate system (zero heading, zero pitch, and zero bank in the  $x'''y'''z'''$  system).

With each rotation of the coordinate axes, the coordinates that specify any point will change. Figure 10 illustrates the relationships between the original coordinate system and the three different primed systems. Figure 10a shows that for any point  $Q(X, Y, Z)$

$$\begin{aligned} X' &= R \cos(n) \\ &= R \cos(m+b) \\ &= R \cos(m) \cos(h) \\ &\quad + R \sin(m) \sin(h) \\ &= X \cos(h) - Y \sin(h) \end{aligned} \quad (1)$$

$$\begin{aligned} Y' &= R \sin(n) \\ &= R \sin(m+h) \\ &= R \sin(m) \cos(h) \\ &\quad + R \cos(m) \sin(h) \\ &= Y \cos(h) + X \sin(h) \end{aligned} \quad (2)$$

$$\text{and} \quad Z' = Z \quad (3)$$

We can see from figure 10b that

$$X'' = X' \quad (4)$$

$$Y'' = Y' \cos(p) + Z' \sin(p) \quad (5)$$

and

$$Z'' = Z' \cos(p) - Y' \sin(p) \quad (6)$$

and we can see from figure 10c that

$$X''' = X'' \cos(b) + Z'' \sin(b) \quad (7)$$

$$Y''' = Y'' \quad (8)$$

and

$$Z''' = Z'' \cos(b) - X'' \sin(b) \quad (9)$$

Substituting (1), (2), and (3) into (4), (5), and (6), and then substituting these results into (7), (8), and (9) yields

$$\begin{aligned} X''' &= [\cos(b)\cos(h) \\ &\quad - \sin(h)\sin(p)\sin(b)] X \\ &\quad + [-\cos(b)\sin(h) \\ &\quad - \sin(p)\cos(h)\sin(b)] Y \\ &\quad + [\cos(p)\sin(b)] Z \end{aligned}$$

$$\begin{aligned} Y''' &= [\sin(h)\cos(p)] X \\ &\quad + [\cos(p)\cos(h)] Y \\ &\quad + [\sin(p)] Z \end{aligned}$$

and

$$\begin{aligned} Z''' &= [-\cos(h)\sin(b) \\ &\quad - \sin(h)\sin(p)\cos(b)] X \\ &\quad + [\sin(h)\sin(b) \\ &\quad - \sin(p)\cos(h)\cos(b)] Y \\ &\quad + [\cos(p)\cos(b)] Z \quad (10) \end{aligned}$$

These equations can be represented using matrix notation, as shown in figure 11. By multiplying out the three matrices, these equations relate

$$\begin{aligned} \begin{bmatrix} X''' \\ Y''' \\ Z''' \end{bmatrix} &= \begin{bmatrix} \cos(b) & 0 & \sin(b) \\ 0 & 1 & 0 \\ -\sin(b) & 0 & \cos(b) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(p) & \sin(p) \\ 0 & -\sin(p) & \cos(p) \end{bmatrix} \begin{bmatrix} \cos(h) & -\sin(h) & 0 \\ \sin(h) & \cos(h) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ &= \begin{bmatrix} \cos(b)\cos(h) - \sin(h)\sin(p)\sin(b) & -\cos(b)\sin(h) - \sin(p)\cos(h)\sin(b) & \cos(p)\sin(b) \\ \sin(h)\cos(p) & \cos(p)\cos(h) & \sin(p) \\ -\cos(h)\sin(p) - \sin(h)\sin(p)\sin(b) & \sin(h)\sin(b) - \sin(p)\cos(h)\cos(b) & \cos(p)\cos(b) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \end{aligned}$$

Figure 11: Equation (10) represented using matrix notation.

the coordinates of a point  $Q$  in the  $xyz$  system to the coordinates of  $Q$  in a system,  $x'''y'''z'''$ , where the observer is in the standard position. Keep in mind that the observer is assumed to be at the origin in the  $xyz$  system prior to these calculations. Of course, the observer is also at the origin of the  $x'''y'''z'''$  system because of the definition of standard position.

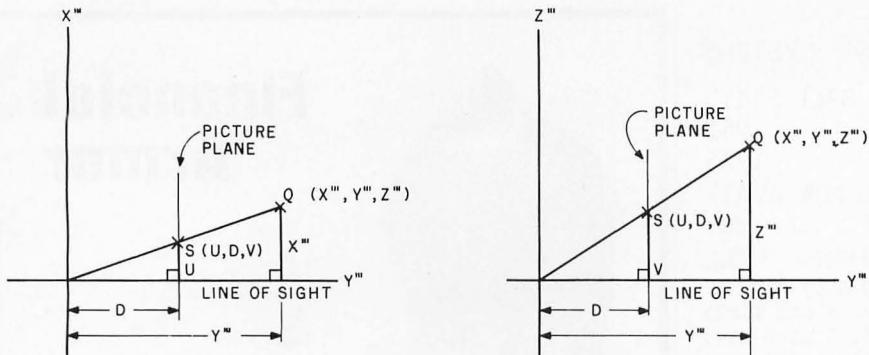
### Projecting into the Picture Plane

Once the observer is in the standard position, it is easy to compute

the perspective of a point. Remember that the perspective of a point  $Q$  is the intersection of the picture plane and the projector line joining the observer and the point. Since the standard position is in use, the observer is located at the origin and the line of sight is the positive  $y'''$  axis.

We have to define a new coordinate system for the picture plane. The two axes in the picture plane are labeled  $u$  and  $v$  such that the  $u$  axis is parallel to the  $x'''$  axis and the  $v$  axis is parallel to the  $z'''$  axis. Thus, the observer interprets the  $u$  axis to be

 <b>IBM PERSONAL COMPUTER</b> 64K MEMORY, 2 DRIVES (320K EACH), COLOR GRAPHICS ADAPTER ..... CALL 64K MEMORY, 2 DRIVES (320K EACH), MONOCHROME DISPLAY & PRINTER ADAPTER .. CALL QUADRAM (64K) ..... 495 VISICALC ..... 160 WORDSTAR ..... 350 <b>MONITORS</b> AMDEX LOW-RES 13" COLOR I ..... 365 AMDEX RGB 13" COLOR II ..... 775 BMC 12" HI-RES GREEN (12EUN) ..... 150 ZENITH 12" GREEN (ZVM-121) ..... 110 NEC 12" GREEN (JB-1201) ..... 169 TAXAN RGB VISION I ..... 350 TAXAN RGB VISION II ..... 525 TAXAN RGB VISION III ..... 600	 <b>OSBORNE</b> CALL FOR PRICE & AVAILABILITY KAYPRO II ..... CALL <b>NEC APC</b> H01 ONE DRIVE B&W SYSTEM .. CALL H02 TWO DRIVES B&W SYSTEM .. CALL H03 TWO DRIVES COLOR SYSTEM ..... CALL <b>SHARP PC-1500</b> PC-1500 HANDHELD COMPUTER .. 255 CE-150 PRINTER/CASSETTE INTERFACE ..... 215 CE-155 8K RAM EXPANSION UNIT .120	 <b>brother HR-1</b> CALL FOR PRICE COMREX CR-1-C ..... CALL OKIDATA MICROLINE 82A ..... CALL OKIDATA MICROLINE 83A ..... CALL C.I.TOH STARWRITER F-10-40PU ..... 1495 C.I.TOH PROWRITER 8510AP ..... 499 EPSON PRINTERS ..... CALL MICROTEK PARALLEL INTERFACE ..... 89 MICROTEK APPLE DUMPLING-GX ..... 129 *IDS 560G PAPER TIGER ..... 1295 *NEC 3530 SPINWRITER ..... 1495 *OLYMPIA ES100 SERIAL ..... 1195 *SPECIAL OF THE MONTH	 <b>S.C.M. SMITH-CORONA TP-1</b> CALL FOR PRICE <b>ATARI 800 &amp; 400</b> ATARI 800 (16K) ..... 675 ATARI 400 (16K) ..... 250 410 PROGRAM RECORDER ..... 75 B10 DISK DRIVE ..... 435 16K RAM MEMORY MODULE ..... 89 850 INTERFACE MODULE ..... 165 830 ACOUSTIC MODEM ..... 159 ATARI VISICALC ..... 199 ATARI WORD PROCESSOR ..... 125 DISKETTES BASF 5 1/4" SS/DD/HUB RING ..... 23 WABASH 5 1/4" SS/DD/HUB RING ..... 22
<b>(800) 854-1941</b> Outside Ca. Order Desk <b>computer age</b> 4688 CONVOY STREET, SAN DIEGO, CA 92111 CALL OR WRITE FOR COMPLETE PRICE LIST		<b>(619) 565-7092</b> Technical & California RISK FREE MAIL ORDER TO ORDER: Please send cashier's check or personal check (allow 10 business days to clear). VISA and Master Card credit card service add 3%. Shipping, handling and insurance in U.S. add 3%. Minimum \$5.00. California residents add 6% sales tax. Foreign orders add 10% for shipping. Equipment is subject to price change and availability. All equipment carries warranty. Store prices may differ from mail order prices.	



**Figure 12:** Projecting a point. Once the observer is in the standard position, it is simple to compute the perspective projection of a point  $Q$ . The perspective projection of a point is the intersection of the point's projector and the picture plane. In this figure,  $D$  is the distance between the observer and the picture plane. See equations (11) through (14) in the text.

horizontal and the  $v$  axis to be vertical. Figure 12 illustrates the relationship between the  $x''''y''''z''''$  coordinates of a three-dimensional point  $P$  and the  $u-v$  coordinates of the point's projection in the picture plane. If  $D$  is the distance between the observer and the picture plane, from similar triangles

$$U = [DX'''']/Y''' \quad (11)$$

and

$$V = [DZ'''']/Y''' \quad (12)$$

When displaying a perspective drawing on a flat screen, it is often convenient to locate the intersection of the viewer's line of sight and the picture plane at the center of the

screen. In this case the  $u$  and  $v$  axes are translated within the plane, and (11) and (12) take the form

$$U = \text{midh} + [\text{ppu } D X'''']/Y''' \quad (13)$$

and

$$V = \text{midv} + [\text{ppu } D Z'''']/Y''' \quad (14)$$

In these equations, ( $\text{midh}$ ,  $\text{midv}$ ) is the center of the viewing screen and  $\text{ppu}$  (pixels per unit) is a scaling factor to compensate for the fact that the unit of measurement of the screen is not the same as the unit of measurement in the coordinate axes.

It is often good to assume that the video monitor is the picture plane. In this case, the distance between the observer and the picture plane,  $D$ , is simply the distance between the computer operator and the terminal's screen. In this configuration, the viewing screen is truly a window through which to view the world. Since a perspective view of a three-dimensional line is always a two-dimensional line, we need only translate, rotate, and project two points for each edge of the scene. The line segment joining the two projected points will accurately represent the edge. However, if one of the endpoints of the segment lies outside the viewing screen, the line segment will have to undergo the process called "clipping." A clipping subroutine just ensures that only the part of the line segment that is within the boundaries of the screen will be drawn.

## The Programs

Listing 1 and listing 2 are, respectively, BASIC and Pascal programs developed for the Apple II computer. The programs use the concepts discussed in this article to develop a perspective view of a three-dimensional object. In these programs, the observer can view the object, centered at the origin, from any point on a celestial sphere surrounding the object. The Pascal version includes capabilities for clipping and, in addition, for colored edges.

Figure 7 illustrates the data structure for both programs. In the BASIC version, the data is stored within the

LINEAR CIRCUITS		THIS MONTH'S SPECIALS		DISC Controllers		REGULATORS		RS232 CONN		74LS SERIES	
TU062 CP - .95	LM061 - 1.75	747 - .50	CA759 - 1.75	1771 ..... \$20.00	340T-5, 6, 8, 9, 12,	LM317T ..... \$1.75	340T-5, 6, 8, 9, 12,	DB 25P male .... \$2.75	74LS05 ..... \$1.35	74LS05 ..... \$1.35	
TU064 CN - 1.10	LM077 - 1.60	CA759 - 1.75	LM310 - 1.75	1791 ..... 29.00	15, 18 or 24V ... \$ .85	78L05 ..... \$ .40	15, 18 or 24V ... \$ .85	DB 25S female .... \$2.75	74LS1412 + 12V 3A \$3.95	74LS1412 + 12V 3A \$3.95	
TL084 - 1.50	LM080 - .95	CA759 - 1.75	LM310 - 1.75	1793 ..... 35.00		78M05 ..... \$ .35		DB 25S female .... \$3.95	74LS240 ..... \$ .90	74LS240 ..... \$ .90	
LM020 - .75	LM021 - .90	1456 - .80	LM307 - 1.75	1795 ..... 45.00		4194D ..... \$ .95		DB 25S female .... \$3.95	74LS241 ..... \$ .90	74LS241 ..... \$ .90	
LM021/700 - .30	LM023 - 2.50	1456 - .80	LM307 - 1.75	1797 ..... 45.00		LM305G ..... \$ .75		DB 25S female .... \$3.95	74LS242 ..... \$ .90	74LS242 ..... \$ .90	
LM031 - .30	LM038 - 1.40	1456 - .80	LM307 - 1.75	D765C ..... 25.00		320T 5, 12, 15 or 24 ... \$ .90		HOODS ..... \$1.25	74LS243 ..... \$ .90	74LS243 ..... \$ .90	
LM008 - .75	LM087 - 1.25	LM201 - .95	LM307 - 1.75						74LS244 ..... \$ .90	74LS244 ..... \$ .90	
LM010 - 1.10	LM093 - 1.00	CA308 - 1.85	LM307 - 1.75						74LS245 ..... \$ .90	74LS245 ..... \$ .90	
LM011 - .50	LM065 - .45	CA308 - 1.85	LM307 - 1.75						74LS246 ..... \$ .90	74LS246 ..... \$ .90	
LM012 - .65	LM066 - .45	CA308 - 1.85	LM307 - 1.75						74LS247 ..... \$ .90	74LS247 ..... \$ .90	
LM024 - .65	LM049 - 3.50	CA308 - 1.85	LM307 - 1.75						74LS248 ..... \$ .90	74LS248 ..... \$ .90	
LM038 - .65	5649 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS249 ..... \$ .90	74LS249 ..... \$ .90	
LM039 - .75	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS250 ..... \$ .90	74LS250 ..... \$ .90	
LM040 - .75	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS251 ..... \$ .90	74LS251 ..... \$ .90	
LM041 - .75	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS252 ..... \$ .90	74LS252 ..... \$ .90	
LM042 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS253 ..... \$ .90	74LS253 ..... \$ .90	
LM043 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS254 ..... \$ .90	74LS254 ..... \$ .90	
LM044 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS255 ..... \$ .90	74LS255 ..... \$ .90	
LM045 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS256 ..... \$ .90	74LS256 ..... \$ .90	
LM046 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS257 ..... \$ .90	74LS257 ..... \$ .90	
LM047 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS258 ..... \$ .90	74LS258 ..... \$ .90	
LM048 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS259 ..... \$ .90	74LS259 ..... \$ .90	
LM049 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS260 ..... \$ .90	74LS260 ..... \$ .90	
LM050 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS261 ..... \$ .90	74LS261 ..... \$ .90	
LM051 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS262 ..... \$ .90	74LS262 ..... \$ .90	
LM052 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS263 ..... \$ .90	74LS263 ..... \$ .90	
LM053 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS264 ..... \$ .90	74LS264 ..... \$ .90	
LM054 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS265 ..... \$ .90	74LS265 ..... \$ .90	
LM055 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS266 ..... \$ .90	74LS266 ..... \$ .90	
LM056 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS267 ..... \$ .90	74LS267 ..... \$ .90	
LM057 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS268 ..... \$ .90	74LS268 ..... \$ .90	
LM058 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS269 ..... \$ .90	74LS269 ..... \$ .90	
LM059 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS270 ..... \$ .90	74LS270 ..... \$ .90	
LM060 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS271 ..... \$ .90	74LS271 ..... \$ .90	
LM061 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS272 ..... \$ .90	74LS272 ..... \$ .90	
LM062 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS273 ..... \$ .90	74LS273 ..... \$ .90	
LM063 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS274 ..... \$ .90	74LS274 ..... \$ .90	
LM064 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS275 ..... \$ .90	74LS275 ..... \$ .90	
LM065 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS276 ..... \$ .90	74LS276 ..... \$ .90	
LM066 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS277 ..... \$ .90	74LS277 ..... \$ .90	
LM067 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS278 ..... \$ .90	74LS278 ..... \$ .90	
LM068 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS279 ..... \$ .90	74LS279 ..... \$ .90	
LM069 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS280 ..... \$ .90	74LS280 ..... \$ .90	
LM070 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS281 ..... \$ .90	74LS281 ..... \$ .90	
LM071 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS282 ..... \$ .90	74LS282 ..... \$ .90	
LM072 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS283 ..... \$ .90	74LS283 ..... \$ .90	
LM073 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS284 ..... \$ .90	74LS284 ..... \$ .90	
LM074 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS285 ..... \$ .90	74LS285 ..... \$ .90	
LM075 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS286 ..... \$ .90	74LS286 ..... \$ .90	
LM076 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS287 ..... \$ .90	74LS287 ..... \$ .90	
LM077 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS288 ..... \$ .90	74LS288 ..... \$ .90	
LM078 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS289 ..... \$ .90	74LS289 ..... \$ .90	
LM079 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS290 ..... \$ .90	74LS290 ..... \$ .90	
LM080 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS291 ..... \$ .90	74LS291 ..... \$ .90	
LM081 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS292 ..... \$ .90	74LS292 ..... \$ .90	
LM082 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS293 ..... \$ .90	74LS293 ..... \$ .90	
LM083 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS294 ..... \$ .90	74LS294 ..... \$ .90	
LM084 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS295 ..... \$ .90	74LS295 ..... \$ .90	
LM085 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS296 ..... \$ .90	74LS296 ..... \$ .90	
LM086 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS297 ..... \$ .90	74LS297 ..... \$ .90	
LM087 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS298 ..... \$ .90	74LS298 ..... \$ .90	
LM088 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS299 ..... \$ .90	74LS299 ..... \$ .90	
LM089 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS300 ..... \$ .90	74LS300 ..... \$ .90	
LM090 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS301 ..... \$ .90	74LS301 ..... \$ .90	
LM091 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS302 ..... \$ .90	74LS302 ..... \$ .90	
LM092 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS303 ..... \$ .90	74LS303 ..... \$ .90	
LM093 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS304 ..... \$ .90	74LS304 ..... \$ .90	
LM094 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS305 ..... \$ .90	74LS305 ..... \$ .90	
LM095 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS306 ..... \$ .90	74LS306 ..... \$ .90	
LM096 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS307 ..... \$ .90	74LS307 ..... \$ .90	
LM097 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS308 ..... \$ .90	74LS308 ..... \$ .90	
LM098 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS309 ..... \$ .90	74LS309 ..... \$ .90	
LM099 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS310 ..... \$ .90	74LS310 ..... \$ .90	
LM100 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS311 ..... \$ .90	74LS311 ..... \$ .90	
LM101 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS312 ..... \$ .90	74LS312 ..... \$ .90	
LM102 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS313 ..... \$ .90	74LS313 ..... \$ .90	
LM103 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS314 ..... \$ .90	74LS314 ..... \$ .90	
LM104 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS315 ..... \$ .90	74LS315 ..... \$ .90	
LM105 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS316 ..... \$ .90	74LS316 ..... \$ .90	
LM106 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS317 ..... \$ .90	74LS317 ..... \$ .90	
LM107 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS318 ..... \$ .90	74LS318 ..... \$ .90	
LM108 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS319 ..... \$ .90	74LS319 ..... \$ .90	
LM109 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75						74LS320 ..... \$ .90	74LS320 ..... \$ .90	
LM110 - .22	5656 - 1.40	CA308 - 1.85	LM307 - 1.75	</td							

**Listing 1:** An Applesoft BASIC program that produces a perspective view of a three-dimensional object.

```

50 REM ---WRITTEN BY ANDREW PICKHOLTZ---
70 REM ---JANUARY 1981---
100 HOME
120 VTAB 9: HTAB 3
140 PRINT "PERSPECTIVE VIEW OF A 3-D OBJECT"
160 PRINT : PRINT
180 PRINT "WRITTEN BY ANDREW PICKHOLTZ - JAN 1981"
200 VTAB 18
220 PRINT " PADDLE #0 CONTROLS THE VIEWER'S PITCH"
240 PRINT "PADDLE #1 CONTROLS THE VIEWER'S HEADING"
280 PRINT : PRINT : HTAB 6: FLASH
300 PRINT "WAIT WHILE LOADING VERTICES"
320 NORMAL : FOR SC = 1 TO 3000: NEXT SC
340 X = Y = Z = X3 = Y3 = Z3 = AM = BM = CM = DM = EM = FM = GM = HM =
IM = D = P = B
= H = U = V = U1 = V1 = 0
360 DIM V(50,3),E(100)
370 REM ---READ DATA---
380 READ NV
400 FOR P = 1 TO NV
420 READ V(P,1),V(P,2),V(P,3)
440 NEXT P
460 READ NE
470 FOR E = 1 TO NE
480 READ E(E)
490 NEXT E
500 HOME : HCOLOR= 0: HPLOT 0,0
510 REM ---COMPUTE OBSERVER'S PARAMETERS---
520 D = 75
540 P = 6.28 * PDL (0) / 255 - 3.1416
560 B = 0
580 H = 6.28 * PDL (1) / 255
600 GOSUB 20100
620 XV = - D * CP * SH: REM ---SEE SUB.---
640 YV = - D * CP * CH
660 ZV = - D * SP
700 REM ---PROJECT NE POINTS---
720 FOR E = 1 TO NE
800 X = V( ABS (E(E)),1)
820 Y = V( ABS (E(E)),2)
840 Z = V( ABS (E(E)),3)
860 GOSUB 20720
900 IF E(E) > 0 THEN HCOLOR= 3: HPLOT U1,V1 TO U,V
920 U1 = U:V1 = V
940 NEXT E
999 REM ---PREPARE FOR NEW FRAME---
1000 VTAB 21: HTAB 8: PRINT "PERSPECTIVE OF A BLOCK"
1010 PRINT "PROGRAM WRITTEN BY ANDREW PICKHOLTZ"
1020 HTAB 8: PRINT "PITCH="; INT (57.2 * P); " HEADING="; INT (57.2 * H)
1060 PRINT "BUTTON #1 TO END-#0 FOR P= , H= ";
1070 VTAB 24: HTAB 27: PRINT " , H= ";
1080 VTAB 24: HTAB 27: PRINT INT (360 * PDL (0) / 255 - 180);
1100 VTAB 24: HTAB 35: PRINT INT (360 * PDL (1) / 255);
1110 REM ---CHECK FOR BUTTON PRESS---
1120 IF PEEK (- 16287) > 127 THEN 500
1140 IF PEEK (- 16286) < 128 THEN 1070
1160 TEXT
1180 END
20000 REM --- 3D PROJECTION SUBS. ---
20100 REM --- SET UP MATRIX ELEMENTS GIVEN PITCH, BANK, HEADING ---
20120 REM --- USE TRANSCENDENTAL FUNCTIONS AS FEW TIMES AS POSSIBLE ---
20140 CH = COS (H):SH = SIN (H)
20160 CP = COS (P):SP = SIN (P)
20180 CB = COS (B):SB = SIN (B)
20200 REM --- SET UP MATRIX ---
20220 AM = CB * CH - SH * SP * SB
20240 BM = - CB * SH - SP * CH * SB
20260 CM = CP * SB
20280 DM = SH * CP
20300 EM = CP * CH
20320 FM = SP
20340 GM = - CH * SB - SH * SP * CB
20360 HM = SH * SB - SP * CH * CB
20380 IM = CP * CB
20400 RETURN

```

program. In the Pascal version, the data must be entered into a disk file. Listing 3 is a simple program that will store the data for the Pascal system.

The first element of data for both programs is the number of vertices that compose the object. Each vertex is then specified by its three coordinates. The unit of measurement in the coordinate system is the centimeter. If you use a monitor that does not measure 12 inches diagonally, you should change the ppu described in equations (13) and (14) to assure a truly accurate representation of distance. The programs assign a number to each vertex that is in the data file; the first vertex is number one, the second number two, and so on.

Following the vertices is the number of lines that are to be projected. This number also indicates how many numbers remain in the file. All of the remaining numbers refer to vertices that are to be projected. If the number is a positive integer, the programs draw a white line from the previously projected vertex to the vertex specified by the integer. The Pascal version will draw colored lines if you append a decimal point and a digit to the integer; point one specifies the color green, point two the color violet, and three and four specify orange and blue, respectively. A negative integer indicates that only the indicated vertex should be plotted, not a line as before.

Both programs have a subroutine (SETUP in the Pascal listing; line 20000 in the BASIC) that computes the elements of matrix in figure 11. You input the pitch and heading of the observer by using the Apple's paddles. The observer's bank is set to zero. Since the observer is on a celestial sphere of radius 75 centimeters, the location of the observer can be computed using a conversion from spherical to rectangular coordinates. Thus:

$$\begin{aligned}
\text{viewer's } x \text{ coordinate} &= -75 \cos(p) \sin(h) \\
\text{viewer's } y \text{ coordinate} &= -75 \cos(p) \cos(h) \\
\text{viewer's } z \text{ coordinate} &= -75 \sin(p)
\end{aligned}$$

*Listing 1 continued on page 492*

*Listing 1 continued:*

```
20600 REM --- SUB. TO TRANSFORM A 3D POINT TO A 2D ---
20620 REM --- X,Y,Z IS THE 3D POINT WHICH IS TRANSFORMED INTO U,V ---
20640 REM --- XV,YV,ZV ARE THE COORDINATES OF THE VIEWER ---
20660 REM --- D IS THE DISTANCE BETWEEN THE OPERATOR'S EYE AND THE SCREEN IN CM
20680 REM --- GOSUB 20100 EVERY TIME THE VIEWERS PITCH, BANK, OR HEADING CHANGES
20700 REM --- TRANSLATE SO THAT VIEWER IS AT THE ORIGIN ---
20720 X = X - XV
20740 Y = Y - YV
20760 Z = Z - ZV
20780 REM --- ROTATE SO THAT THE VIEWER IS LOOKING DOWN THE Y-AXIS ---
20800 X3 = AM * X + BM * Y + CM * Z
20820 Y3 = DM * X + EM * Y + FM * Z
20840 Z3 = GM * X + HM * Y + IM * Z
20860 REM ---PROJECT INTO 2D SCREEN---
20880 U = 135 + 13.5 * D * X3 / Y3
20900 V = 80 - 11.5 * D * Z3 / Y3
20920 RETURN
30000 REM ---- NUMBER OF VERTICES---
30020 DATA 8
30040 REM ---- VERTEX COORDINATES---
30060 DATA 5.25,-2,3.25
30080 DATA -5.25,-2,3.25
30100 DATA -5.25,-2,-3.25
30120 DATA 5.25,-2,-3.25
30140 DATA 5.25,2,3.25
30160 DATA -5.25,2,3.25
30180 DATA -5.25,2,-3.25
30200 DATA 5.25,2,-3.25
31000 REM ---- NUMBER OF EDGES---
31020 DATA 16
31040 REM ---- EDGES ---
31060 REM ---NEG. EDGES START NEW CURVE---
31080 DATA -1,2,3,4,1,5,6,7,8,5
31100 DATA -2,6,-3,7,-4,8
```

Note that the distance between the observer and the picture plane (DIS or D) is equal to the distance between the observer and the center of the object (the radius of the sphere). Therefore, the dimensions of the perspective of the object will approximate those of the object itself.

The second subroutine in each program (PROJECT, or line 20600) does the computations needed to project each point. Thus, each subroutine first transforms the point into the triple-primed system. Then, each projects the point into the picture plane using formulas (13) and (14). The program has to call this second subroutine every time a point is projected. The program has to call the first subroutine only when the viewing parameters change.

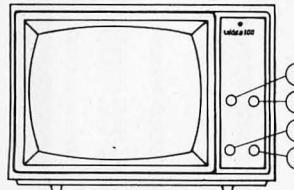
### Using the Programs

The BASIC program in listing 1 is ready to run as is. Lines 30000-31100 of listing 1 contain the data for a rect-

*Text continued on page 500*

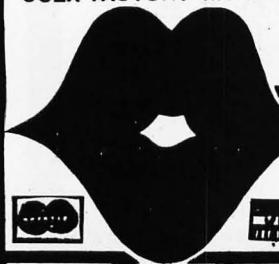
## 'DO YOU KNOW WHERE YOUR COMPUTER IS TONITE?'

by AMDEK



**VIDEO 100**  
12" BLACK&WHITE  
MONITOR  
FULL FACTORY WARRANTY  
**\$79.95**

for **APPLE**  
**16K RAM CARD**  
\*LANGUAGE TRANSPARENT  
**\$69.95**  
COEX FACTORY WARRANTY



### floppy DISKETTES

5 1/4" - 100 PER BOX  
ALL CERTIFIED - 100% GUARANTY

**\$149.00**

ABOVE WITH HUB RINGS  
**\$169.00**

### VISION-80® 80x24 Video

Display Card **\$249<sup>00</sup>**

Vista Computer Company's new Vision-80 board is a sophisticated yet easy to use video display card for the Apple™ computer.

### FLOPPY DISK DRIVE

DESIGNED FOR  
YOUR APPLE®

**\$287<sup>95</sup>**

TRACK ZERO MICRO SWITCH

DOS 3.2.1 & DOS 3.3

PASCAL & CP/M

Fourth  
Dimension  
Systems

CONTROLLER CARD FOR ABOVE **\$99.00**

### EXTENDER CARDS

FOR APPLES... **\$16.95**

FOR I.B.M.... **\$19.95**

### PROTOTYPING CARDS

**Apple \$19.95**

**I BM \$49.95**

NEW FROM COEX

### EPSON TO APPLE

### PARALLEL INTERFACE!

**\$49.95**

CABLE INCLUDED

# Components Express, INC

1380 E. Edinger, Santa Ana, CA 92705 (714) 558-3972

**Listing 2: Object3d**, an Apple Pascal program that produces a perspective view of a three-dimensional object.

```

(*$5+*) (* SWAPPING OPTION *)
PROGRAM OBJECT3D;
DIS : REAL;
DONE : BOOLEAN;

PROCEDURE TITLE;
BEGIN
  PAGE ('OUTPUT');
  GOTOXY (3,6);
  WRITELN ('PERSPECTIVE VIEW OF A 3-D OBJECT');
  WRITELN ('WRITTEN BY ANDREW PICKHOLTZ - JAN 1981');
  GOTOXY (1,16);
  WRITELN ('#0 CONTROLS OBSERVER'S PITCH');
  WRITELN ('#1 CONTROLS OBSERVER'S HEADING');
  WRITELN ('#2 TO BE DISPLAYED');
  WRITELN ('#3 TO BE DISPLAYED');
END;

TYPE
  RANGVER = 1..MAXVER;
  RANGEIG = 1..MAXENG;
  POINT3D = ARRAY [X,Y,Z] OF REAL;
  POINT2D = ARRAY [U,V] OF REAL;
  MANY3D = ARRAY [RANGVER] OF POINT3D;

PROCEDURE READDATA;
VAR I : RANGVER;
  J : RANGEIG;
  FUNCTION LOAD:REAL;
BEGIN
  LOAD := DATAFILE^;
  GET (DATAFILE);
END;
BEGIN
  READLN (OBJECTNAME);
  RESET (DATAFILE,OBJECTNAME);
  (* LOAD VERTICES *)
  NUMVER := TRUNC (LOAD);
  FOR I:= 1 TO NUMVER DO
    BEGIN
      VERTEX : ARRAY [RANGEIG] OF REAL;
      DATAFILE : FILE OF REAL;
      OBJCTNAME : STRING;
      OBSCOOR : POINT3D; (*OBSERVER'S COORDINATES*)
      P,B,H : REAL;
      SP,CP,SB,CB,SH,CH : REAL;
      AM,BM,CM,EM,FM,GM,HM,IM : REAL;
      NUMVER : RANGVER;
      NUMENG : RANGEIG;
    END;
  END;
END;

```

Listing 2 continued:

```
VERTEX [I,Y] := LOAD;
VERTEX [I,Z] := LOAD;
END; (* LOAD EDGES *)
NUMEDG := TRUNC (LOAD);
FOR J:= 1 TO NUMEDG DO
BEGIN
  ENGE [J,J] := LOAD;
END;
CLOSE (DATAFILE);
END; (*READDATA*)

PROCEDURE SETUP;
CONST EPSILON = 2.5E-2;
BEGIN
  (* PITCH, BANK, HEADING, DISTANCE *)
  DIS:=75;
P:=2*PI*FAINLE(0)/255-PI;
IF ABS(P) < EPSILON THEN P:=0;
B:=0;
H:=-2*PI*FAINLE(1)/255;
(* TRANSCENDENTAL FUNCTIONS *)
SP:=SINK(F); CF:=COS(F);
SR:=SINK(B); CR:=COS(B);
SH:=SINK(H); CH:=COS(H);
(* MATRIX COMPONENTS *)
AM := CH*CB-SH*SF*SB;
BM := -SH*CB-CH*SF*SB;
CM := CP*SR;
DM := SH*CP;
EM := CP*CH;
FM := SP;
GN := -CH*SB-SH*SF*CB;
HM := SH*SB-CH*SF*CB;
IM := CP*CB;
(* OBSERVER'S COORDINATES *)
OBSCOORI X] := -DIS*CP*SH;
OBSCOORI Y] := -DIS*CP*CH;
OBSCOORI Z] := -DIS*SF;
END;
```

```
PROCEDURE PROJECT (PT3D : POINT3D; VAR PT2D : POINT2D);
VAR ROTPT : POINT3D;
BEGIN
  (* TRANSLATE SO THAT OBSERVER IS AT ORIGIN *)
  PT3DCX] := PT3D[0] - OBSCOORI X];
  PT3DCY] := PT3D[1] - OBSCOORI Y];
  PT3DCZ] := PT3D[2] - OBSCOORI Z];
  (* ROTATE SO THAT OBSERVER IS LOOKING DOWN Y-AXIS *)
  ROTPT[X] := AM*PT3DCX] + BM*PT3DCY] + CM*PT3DCZ];
  ROTPT[Y] := DM*PT3DCX] + EM*PT3DCY] + FM*PT3DCZ];
  ROTPT[Z] := GM*PT3DCX] + HM*PT3DCY] + IM*PT3DCZ];
  (* PROJECT INTO PICTURE PLANE AT DISTANCE DIS *)
  PT2D [U] := MIDH + FFCMH*DIS*ROTPT[X]/ROTPTY];
  PT2D [V] := MIDV + FFCMV*DIS*ROTPT[Z]/ROTPTY];
END;

PROCEDURE DRAW;
VAR
  J : RANGEID;
  SCRNP1 : POINT2D;
BEGIN
  SETUP;
  FOR J := 1 TO NUMEDG DO
    BEGIN
      (* DELETE FRACTIONAL PART AND PROJECT *)
      PROJECT ( VERTEX(ABS(TRUNC(EDGE [JJ]))), SCRNP1 );
      (* IF EDGE IS NEG THEN LINE IS NOT DISPLAYED *)
      IF EDGE [JJ] < 0
        THEN PENCOLOR (NONE)
        (* FRACTIONAL PART OF EDGE DETERMINES COLOR *)
      ELSE
        CASE TRUNC(10*(EDGE [JJ]-TRUNC(EDGE [JJ])) GF
          0 : PENCOLOR (WHITE);
          1 : PENCOLOR (GREEN);
          2 : PENCOLOR (VIOLET);
          3 : PENCOLOR (ORANGE);
          4 : PENCOLOR (BLUE);
        END;
      MOVETO (ROUND(SCRNP1[U]), ROUND(SCRNP1[V]));
    END; (*DRAW*)
  END;
```

Listing 2 continued on page 498

*Listing 2 continued:*

```
BEGIN
  TITLE;
  READDATA;
  INITTURTLE;
  REPEAT
    DRAW;
    NEWFRAME;
    UNTIL DONE;
    PAGE (OUTPUT);
    TEXTMODE;
    WRITELN ('NONE...');
  END.
```

```
  (* COMMENTS AROUND OBJECT *)
  PENCOLOR (NONE);
  MOVETO (15,183);
  WSTRING ('PERSPECTIVE VIEW OF A ');
  WSTRING (OBJECTNAME);
  MOVETO (10,174);
  STR (ROUND (F*RAADIEG),S);
  WSTRING ('PITCH = ');
  WSTRING (S);
  MOVETO (180,174);
  STR (ROUND (H*RAADIEG),S);
  WSTRING ('HEADING = ');
  WSTRING (S);
  MOVETO (10,2);
  WSTRING ('BUTTON #1 ENIG - #0 FOR ');
  WSTRING ('UPDATE COMMENTS - CHECK BUTTONS ')
REPEAT
```

```
  MOVETO (178,2);
  WSTRING ('');
  MOVETO (178,2);
  PDL := PADULE(0)/255*360-180;
  STR (ROUND (PDL),S);
  WSTRING ('P=');
  WSTRING (S);
  MOVETO (227,2);
  PDL := PADULE(1)/255*360;
  STR (ROUND (PDL),S);
  WSTRING ('H=');
  WSTRING (S);
  ADVANCE := BUTTON (0);
  DONE := BUTTON (1);
  UNTIL ADVANCE OR DONE;
  FILSCREEN (BLACK);
END;
```

*Listing 3: A Pascal program that stores data for use by Object3d, the program given in listing 2.*

```
PROGRAM FILEWRITE;
(* PROGRAM TO INSERT DATA INTO A FILE *)
VAR
  FILENAME : STRING;
  DISK : FILE OF REAL;
  NUMBER : REAL;
BEGIN
  WRITELN;
  WRITE (' NAME OF OBJECT <FILE> ');
  READLN (FILENAME);
  WRITELN ('< CNTL-C > TO STOP');
  WRITELN;
  REWRITE (DISK,FILENAME);
  REPEAT
    READLN (NUMBER);
    DISK1 := NUMBER;
    PUT (DISK);
    UNTIL EOF;
  CLOSE (DISK,LOCK);
END.
```

**Listing 4:** Changes for the BASIC program in listing 1 that will produce a perspective of a dodecahedron.

```

]LIST
430 V(P,1) = 6 * V(P,1):V(P,2) = 6 * V(P,2):V(P,3) = 6 * V(P,3)
1000 VTAB 21: HTAB 4: PRINT "PERSPECTIVE OF A DODECAHEDRON"
30000 REM ---NUMBER OF VERTICES---
30020 DATA 20
30040 REM ---VERTEX COORDINATES---
30060 DATA 0,-.3568,.9342,.5774,-.5774,.5774
30080 DATA .9342,0,.3568,.5774,.5774,.5774
30100 DATA 0,.3568,.9342,-.5774,.5774,.5774
30120 DATA -.9342,0,.3568,-.5774,-.5774,.5774
30140 DATA -.3568,-.9342,0,.3568,-.9342,0
30160 DATA .3568,.9342,0,-.3568,.9342,0
30180 DATA -.5774,-.5774,-.5774,.9342,0,-.3568
30200 DATA .5774,-.5774,-.5774,0,.3568,-.9342
30220 DATA .5774,.5774,-.5774,0,.3568,-.9342
30240 DATA -.5774,.5774,-.5774,-.9342,0,-.3568
31000 REM --- NUMBER OF EDGES---
31020 DATA 40
31040 REM --- EDGES ---
31060 REM ---NEG. EDGES START NEW CURVE---
31080 DATA -1,2,3,4,5,6,7,8,1,5
31100 DATA -14,15,16,17,18,19,20,13,14,18
31120 DATA -8,9,10,2
31140 DATA -13,9
31160 DATA -15,10
31180 DATA -4,11,12,6
31200 DATA -17,11
31220 DATA -19,12
31240 DATA -16,3
31260 DATA -7,20
31280 END

```

**Listing 5:** A Pascal program that displays an accurately proportioned model of a DNA molecule.

```

PROGRAM MAKEINA;
(* PROGRAM TO COMPUTE DOUBLE-HELIX *)
(* ANDREW PICKHOLTZ - JAN 1981 *)

USES APPLESTUFF,TRANSCEND;

CONST
  (* DESCRIPTION OF DOUBLE-HELIX *)
  (* SEE J.D.WATSON, MOLECULAR BIOLOGY OF THE GENE *)
  (* SECOND EDITION, PAGES 261-262 *)
  (* ALL UNITS IN ANGSTROMS *)
  RADIUS = 10;
  HTURN = 34;      (* HEIGHT OF ONE TURN *)
  NNPT = 10;       (* NUMBER OF NUCLEOTIDES PER TURN *)
  OFFSET = 2.402;  (* OFFSET FOR SECOND HELIX IN RADIANS *)
  PI = 3.1416;
  CMPA = 0.1569;  (* CM PER ANGSTROM FOR MAGNIFICATION *)
  NNUCL = 15;      (* HALF NUMBER OF NUCLEOTIDES TO DISPLAY *)

VAR
  NUCL : -NNUCL..NNUCL;
  I,J : INTEGER;
  DISK : FILE OF REAL;
  COLOR : REAL;

```

*Listing 5 continued on page 502*

*Text continued from page 492:*

angular parallelepiped (a prism whose bases are parallelograms), i.e., a block. Listing 4 has the changes that should be made to project a dodecahedron (a solid with 12 faces) instead of a block. To produce a full-screen dodecahedron, you must multiply the data elements in lines 30060-30240 by 6. Once compiled, the Pascal program in listing 2 will display any object that is represented in a disk file. The program in listing 3 will load data into a file.

Objects can also be created by the computer. For example, listing 5 contains a Pascal program that will create an accurately proportioned model of a DNA molecule. The double helix is represented by 62 vertices connected by more than 100 straight lines.

Photo 1 is an actual photograph of a real dodecahedron. Photos 2 through 6 show examples of the use of the programs. Photo 2 displays computer-drawn perspectives from various viewing positions. Similarly, photo 3 shows the wire-frame perspective of a block. Photo 4 shows drawings of an object that you are more likely to encounter, a house. You can see that these photos exhibit the properties of true perspective that were shown in figure 4. The DNA molecule shown in photo 5 is even more interesting. The orientation of the double helix displayed in photo 5a is like those found in many textbooks.

Pascal draws an object significantly faster than does BASIC. On the system used, however, the speed at which the object is drawn and the screen erased is not nearly fast enough to produce the appearance of smooth motion, let alone freedom from flickering (this would require a minimum of 30 frames per second). Thus, although the programs have the inherent capability of displaying a continuous change in view of the object, either a faster machine or a more efficient compiler, or both, would be necessary to achieve this result.

#### Extensions, Anyone?

The most obvious extension to wire-frame perspective is to remove from the display the lines and sur-

**TI**

## "SILENT 745" Terminals

Over 200 terminals in excellent condition (light commercial use) under continuous Texas Instruments maintenance. Available immediately—free delivery. Call collect for more information.

**Russ White**  
**Multi List Inc.**  
**(212) 997-6075**

*Listing 5 continued:*

```
PROCEDURE DUMP (DATA : REAL);
BEGIN
  DISK↑ := DATA;
  PUT (DISK);
END;

BEGIN
  WRITELN;
  WRITELN ('WRITING');
  REWRITE (DISK,'DOUBLE-HELIX');
  (* NUMBER OF POINTS *)
  DUMP (2*(2*NNUCL+1));
  FOR NUCL := -NNUCL TO NNUCL DO
    BEGIN
      (* FIRST HELIX *)
      DUMP (CMPA*RADIUS* COS(NUCL*2*PI/NNPT));
      DUMP (CMPA*RADIUS* SIN(NUCL*2*PI/NNPT));
      DUMP (CMPA*HTURN/NNPT*NUCL);
      (* SECOND HELIX *)
      DUMP (CMPA*RADIUS* COS(NUCL*2*PI/NNPT+OFFSET));
      DUMP (CMPA*RADIUS* SIN(NUCL*2*PI/NNPT+OFFSET));
      DUMP (CMPA*HTURN/NNPT*NUCL);
    END;
    (* NUMBER OF EDGES *)
    DUMP (8*NNUCL);
    FOR J := 1 TO 2*NNUCL DO
      BEGIN
        I := 2*j-1;
        DUMP (-I);
        DUMP (I+2);
        COLOR := ((RANDOM MOD 4)+1)/10;
        DUMP (I+3+COLOR);
        DUMP (I+1);
      END;
    CLOSE (DISK,LOCK);
    WRITELN ('DONE...');
  END.
```

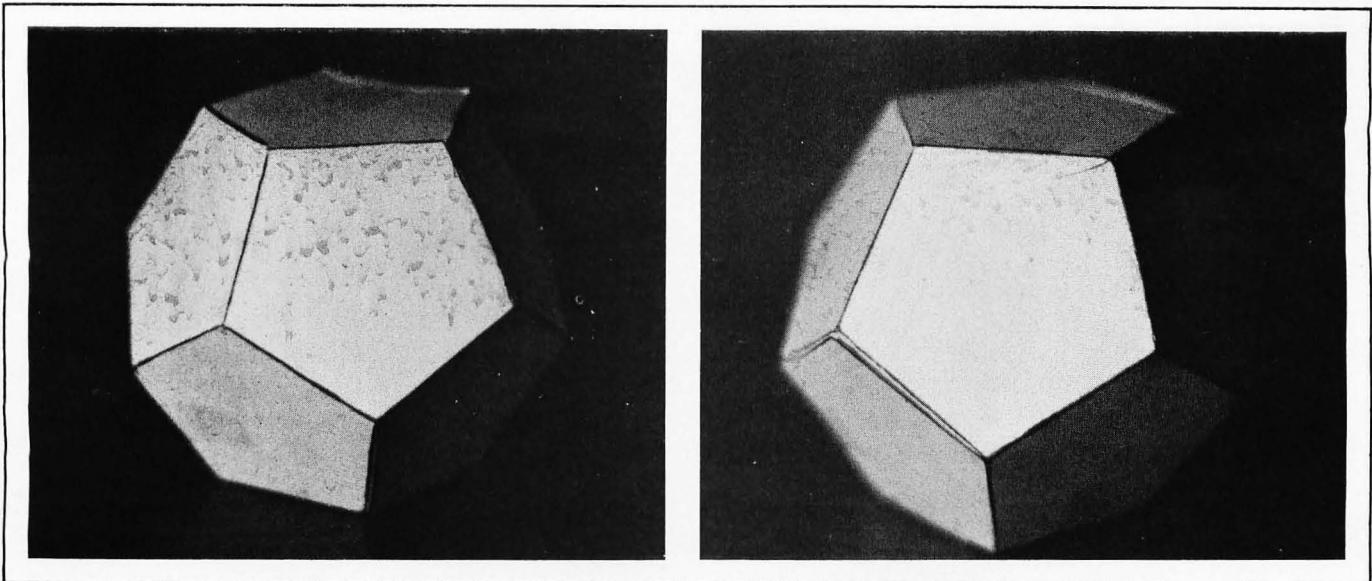
faces that should be hidden, as shown in photo 6. A first step to accomplish this task of hidden-line removal is to define the faces of the object. We can do that by discarding the data structure represented in figure 7 in favor of the data structure in figure 6. The next step in hidden-line removal is to determine which faces shield the others.

Many methods of hidden-line removal are available. All require much more computer time than the wire-frame representation. One procedure, the depth-buffer algorithm, requires that the depth of every pixel on the screen be recorded. Before drawing a new point into the screen, the depth of the point to be displayed

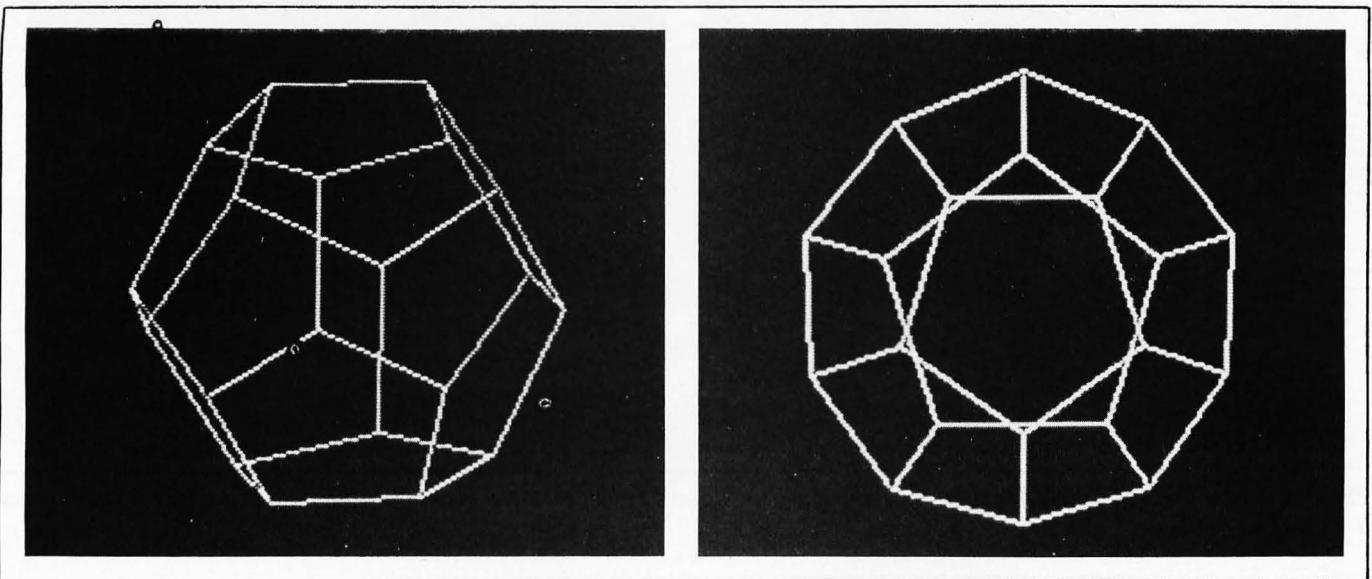
is compared with the depth of the existing pixel. The new point will be drawn only if it is closer to the observer than the existing screen point. Although this algorithm is relatively simple, it requires an enormous amount of computer memory.

Another method of hidden-line removal, the priority algorithm, requires that all the faces be sorted in the order that they are to be drawn into the screen. Thus, the faces in the foreground will block the faces behind them, since the foreground faces will be drawn last. One drawback of the priority algorithm is that it cannot draw cyclically overlapping polygons.

Wire-frame perspective could also



**Photo 1:** A real regular dodecahedron.



**Photo 2:** A screen representation of a wire-frame dodecahedron. The video monitor was connected to an Apple II.

be extended to draw curves rather than just straight lines. The Bezier and B-spline methods are among the many techniques for interpolating curves with a finite number of points.

The interposition of two perspective views of an object from slightly different positions can create the effect of stereo vision. One view could be drawn on the left-hand side of the screen and the other on the right-hand side. Alternatively, an anaglyph (a stereoscopic picture using two different colors, similar to three-dimensional movies now being shown on TV) can be produced by drawing one view in one color and

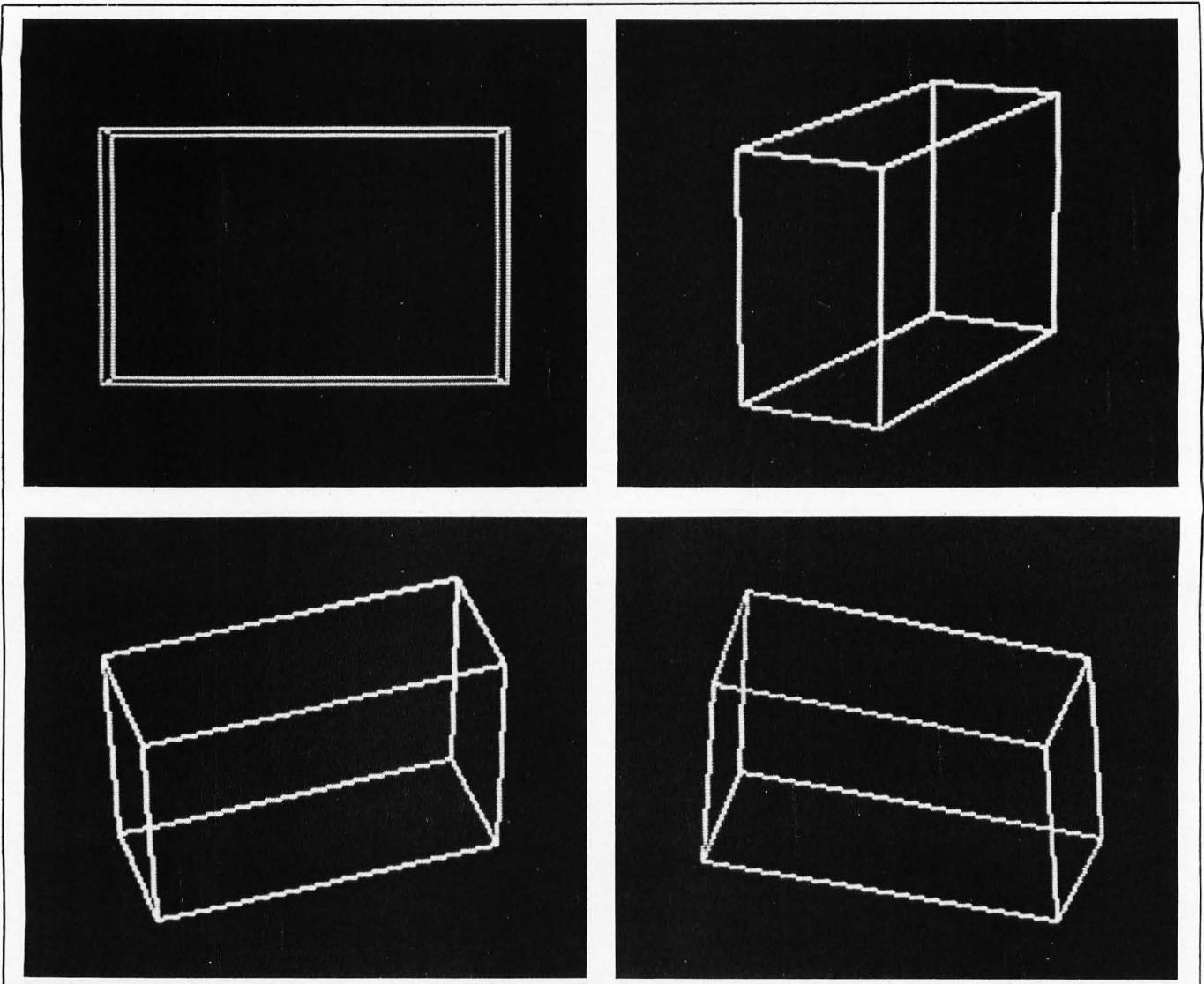
the other slightly displaced view in another color. Of course, viewing glasses with correspondingly colored filters would be required to perceive the stereo effect of the anaglyph.

Many more difficult problems present themselves when searching for further extensions to the wire-frame perspective. Extraordinary realism in three-dimensional graphics can be achieved by including more of the physical characteristics of real objects, such as shading, shadowing, texture, reflectivity, and transparency. These characteristics could all improve the realism of a computer-drawn scene. Although an

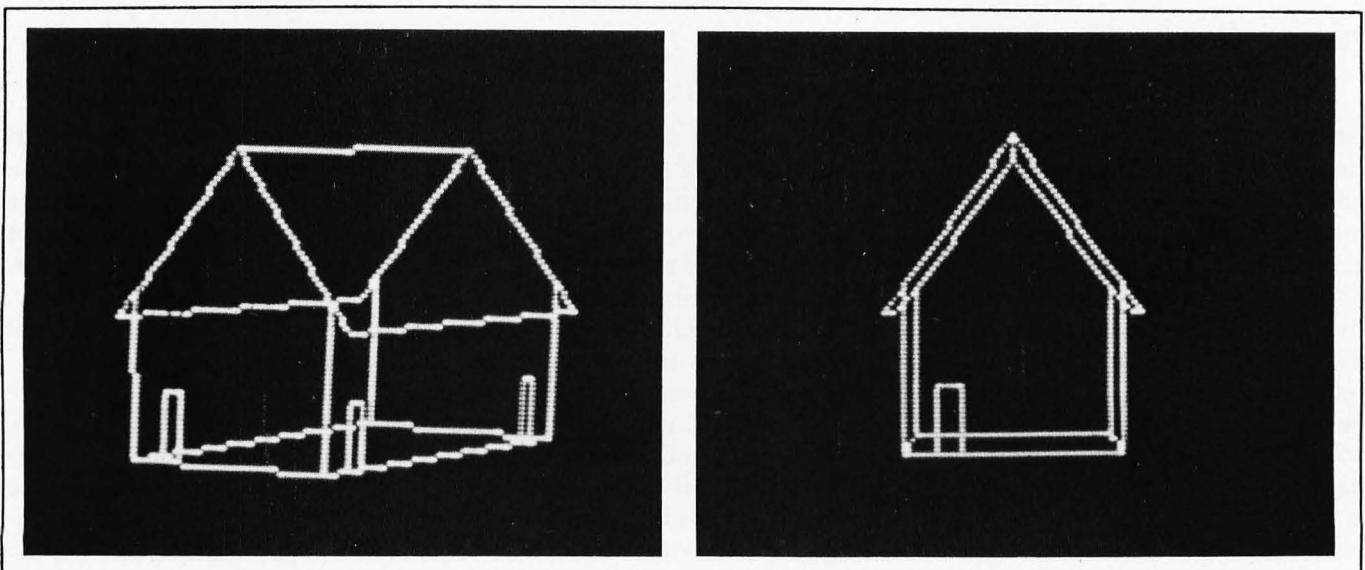
artist could produce a painting with all these characteristics, the computer-drawn scene could be manipulated interactively to present various alterations to the observer, such as different viewing angles, changes in scale, and even topological transformations of the scene.

### Conclusion

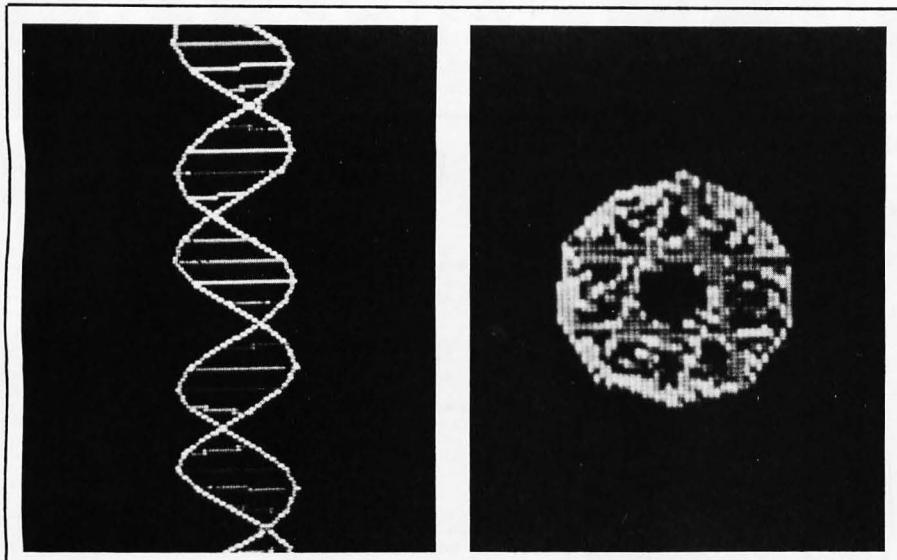
The new generation of microcomputers that is now entering the marketplace will provide an abundance of opportunities for writing and viewing computer graphics. More powerful processors, higher-resolution monitors, and greater



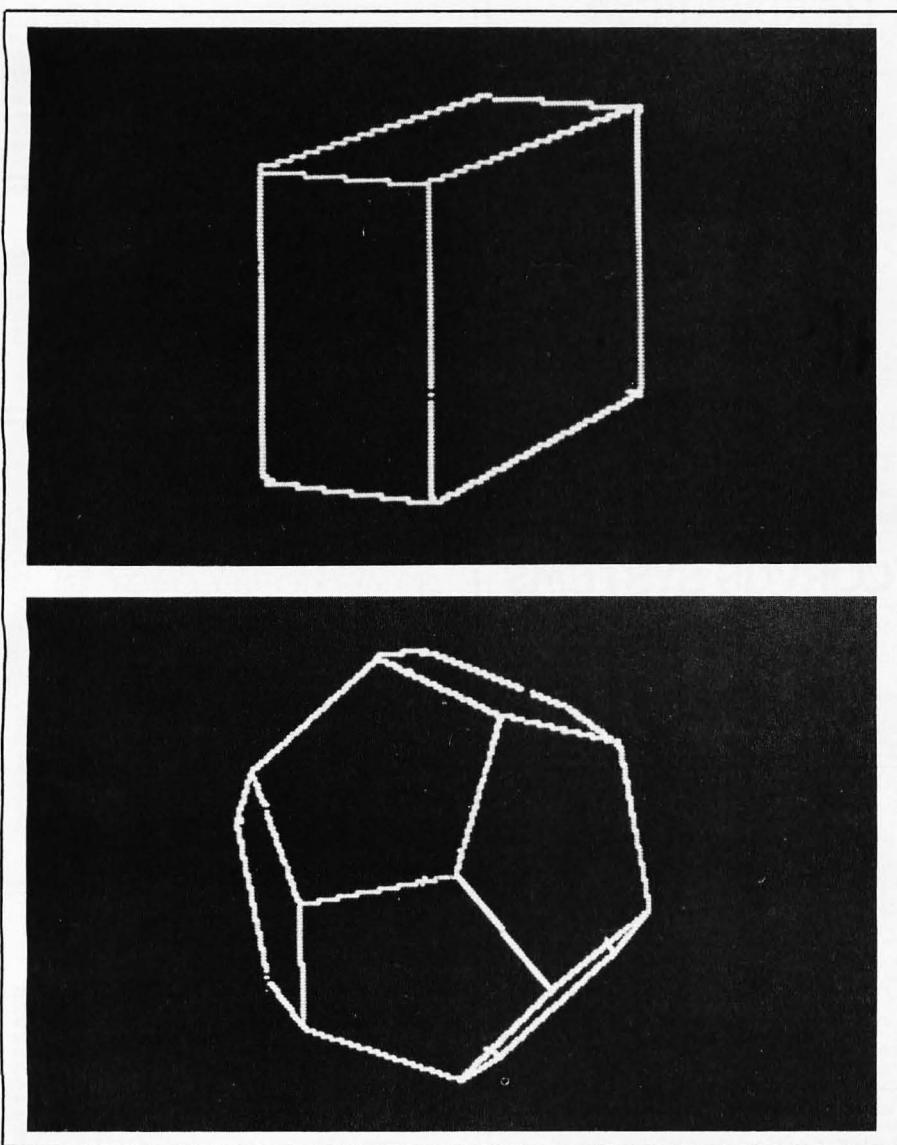
**Photo 3:** Perspective view of a block from a variety of angles. Negative pitch produces views from above, and positive pitch produces a view from below.



**Photo 4:** Perspective view of a wire-frame house.



**Photo 5:** Perspective view of a double helix. The observer's line of sight is perpendicular to the axis of the double helix, and then down the axis of the double helix.



**Photo 6:** Perspective of a block and a dodecahedron with lines that would be hidden by opaque surfaces removed from the representation (photos were manually produced).

memory-addressing capacity will enable programmers to use some of the techniques that were impossible on an Apple II. The same improvements will make computer graphics more exciting to both sophisticated and naive observers. I hope that this article not only interests today's Apple II owners, but also encourages them and others to write software that exploits the impressive graphics capabilities of the new machines. ■

#### References

- Albert, A. Adrian. *Solid Analytic Geometry*. Chicago: University of Chicago Press, 1966.
- Artwick, Bruce A. *Three Dimensional Microcomputer Graphics*. Culver City, CA: Sublogic Company, 1977.
- Chasen, Sylvan H. *Geometric Principles and Procedures for Computer Graphics Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1978.
- Foley, James and Andries van Dam. *Fundamentals of Interactive Computer Graphics*. Reading, MA: Addison-Wesley, 1981.
- Giloil, Wolfgang K. *Interactive Computer Graphics: Data Structures, Algorithms, Languages*. Englewood Cliffs, NJ: Prentice-Hall, 1978.
- Greco, Simon. *The Art of Perspective Drawing*. New York: M. Grumbacher, 1975.
- Horn, Berthold K. P. "Hill Shading and the Reflectance Map." *Proceedings of the IEEE*, January 1981.
- Hungerford, Joel C. "Graphic Manipulations Using Matrices." *BYTE*, September 1978, page 156.
- Image, Object, and Illusion (A Scientific American Publication)*. San Francisco: W. H. Freeman and Company, 1974.
- Langridge, Robert, Thomas E. Ferrin, Irwin D. Kuntz, and Michael L. Connolly. "Real-Time Color Graphics in Studies of Molecular Interactions." *Science*, February 13, 1981.
- Lerner, Eric J. "The Computer Graphics Revolution." *Spectrum*, February 1981.
- Newman, William M. and Robert F. Sproull. *Principles of Interactive Computer Graphics*, second ed. New York: McGraw-Hill, 1979.
- Pearce, Peter and Susan Pearce. *Polyhedra Primer*. New York: Van Nostrand Reinhold, 1978.
- Potsdamer, Jeffrey L. "The Mathematics of Computer Graphics." *BYTE*, September 1978, page 22.
- Requicha, Aristides A. G. "Representations for Rigid Solids: Theory, Methods, and Systems." *ACM Computing Surveys*, December 1980.